A propos de l'Enseignement de l'Informatique Musicale à l'Université

Jacques Arveiller

Communication présentée à l'UNESCO Workshop on Computer Music,
Institute of Musicology, Université d'Aarhus, Danemark, le 1 Septembre 1978.

Artinfo/Musinfo n°30
1980

Un compositeur de musique se réfère toujours dans sa pratique à d'autres compositeurs, du passé et du présent(1). L'idée-même de composition musicale est indissociable de la question de sa transmission pédagogique, avec tout ce que cela comporte d'aspects psychologiques et institutionnels. Car une partie seulement d'un savoir compositionnel est susceptible d'être écrite, sous forme de traité, de manuel, de méthode.

Avec l'entrée en scène des ordinateurs, si la question de fond reste la même, la didactique de la composition va présenter quelques particularités et va achopper sur quelques difficultés spécifiques.

Dans la mesure où, de plus en plus, l'ordinateur va interférer dans la composition musicale, il m'apparaît profitable de relater ici une expérience qui se déroule depuis 1970, et d'indiquer quelques solutions que nous avons tenté de donner à ces difficultés.

Rappelons tout d'abord pour mémoire quelques particularités de la pratique musicale avec ordinateur. Deux points semblent important dans la mesure où ils font surgir des problèmes nouveaux, justiciables de solutions pédagogiques nouvelles.

D'une part la composition musicale avec ordinateur suppose la pratique simultanée de deux activités créatrices qui se chevauchent, avec le risque constant que l'une prenne le pas sur l'autre. Je considère en effet la programmation comme une activité créatrice très complète, qui peut dans certains cas se suffire à elle-même.

D'autre part on assiste à une fusion des fonctions musicales traditionnelles. Au-delà de l'indistinction compositeur/exécutant telle qu'elle se présente par exemple dans l'improvisation, l'informatique musicale instaure dans bien des cas une indistinction entre composition, exécution et lutherie, ce qui n'est pas sans conséquences au niveau des étudiants.

---

(1) Je tiens à remercier ici Patrick Greussay, qui m'a confié un enseignement dont il avait auparavant la charge. Je lui suis redevable, directement ou indirectement, des quelques idées exprimées dans cette communication.

Un second ordre de particularités est lié à l'implantation universitaire. L'informatique musicale ne peut en effet se pratiquer que là où se trouvent des ordinateurs, et là où l'on est prêt à mettre en jeu ces ordinateurs pour faire de la musique. La musique réalisée avec ordinateur est encore une des plus chères qui soient, du point de vue du coût de production. C'est dire que pour le débutant, le lieu privilégié de l'apprentissage est actuellement encore l'université.

L'Université de Paris 8-Vincennes regroupe 32500 étudiants dont 900 en Musique et 1700 en Informatique. Les cours d'informatique musicale sont suivis principalement par les étudiants de ces deux départements, auxquels s'ajoutent des étudiants venus d'autres départements, et un certain nombre d'auditeurs libres (principalement musiciens professionnels venus apprendre pour leur propre compte une technique nouvelle). L'ouverture des cours à toute personne désireuse d'y assister a donc pour conséquence une grande hétérogénéité au niveau du bagage préalable.

Vincennes est une université à vocation plus littéraire que scientifique, donc munie de moyens de calcul relativement limités (Télémécanique T1600 et terminal de PDP10 en télétraitement). L'accent y est donc mis d'une part sur la construction de *software*, d'autre part sur les microprocesseurs et leurs applications. Parmi les réalisations récentes orientées sur les applications artistiques à la disposition des étudiants, on note ainsi par exemple un système LISP sur microprocesseur (VLISP 8 de J. Chailloux), une télévision en couleurs gérée point par point par microprocesseur (COLORIX de L. Audoire) ou le Système Portable de Synthèse Hybride permettant le contrôle de synthétiseurs de son (fruit d'une collaboration : D. Roncin, C. Colère, P. Greussay, etc.).

L'Université de Vincennes a été créée dans une optique pluridisciplinaire. Par voie de conséquence s'y est toujours développée une collaboration de tous les instants entre les différents départements (Informatique, Musique, Arts Plastiques). Et du même coup un travail collectif entre musiciens et artistes plasticiens.

Il faut dire que le Département d'Informatique lui-même a été créé en 1969 par des peintres et musiciens à la recherche d'un nouvel outil de création. L'optique vincennoise est donc un peu particulière : si l'on enseigne toutes les branches de l'informatique, l'idée règne que faire du son ou des images est une des meilleures choses qu'on puisse faire avec un ordinateur, et non pas une activité un peu superflue ou vaguement honteuse. Les machines y sont donc en libre-service, et pas seulement à certaines heures.

Comme dernière particularité de Vincennes, il faut aussi signaler l'existence du Groupe Art et Informatique de Vincennes (GAIV). Ouvert sur l'extérieur, puisqu'il s'occupe de la production de concerts, d'expositions, de spectacles et de conférences, il regroupe des artistes qui sont soit enseignants soit étudiants. Son articulation avec l'université est souple : indépendance sur le plan administratif, préservant cependant la possibilité d'utilisation extérieure d'un matériel appartenant à l'université.

Une université est par essence un lieu de passage. Le GAIV en est un lui aussi. Il ne tire pas sa cohérence d'une direction d'ensemble ou d'un projet collectif (auquel il conviendrait de se soumettre) mais d'une référence commune à un même outil technologique

(par exemple le microprocesseur) et à une même méthodologie (comme la référence à l'Intelligence Artificielle).

Si le *software* est le plus souvent commun, la trace musicale est toujours personnelle, et l'on s'y défie de l'orthodoxie esthétique. Les musiques produites avec l'ordinateur à Vincennes sont donc très diverses dans leur style et leurs références.

★
★ ★

Un troisième et dernier ordre de particularité est lié à la pédagogie. Trois cours annuels (neuf heures par semaine) sont spécifiquement dévolus à l'informatique musicale : Musique et Informatique (J. Arveiller), Synthèse Musicale Hybride (G. Dalmasso) et Formalisation Informatique de Structures Musicales (A. Riotte). S'y ajoutent la part informatique de certains enseignements de musique (M. Battier, G.G. Englert) et la part de la musique dans certains enseignements d'informatique (P. Greussay) On n'oublie pas non plus ce qu'apportent les conférenciers extérieurs.

Sans vouloir subsumer la diversité vincennoise, les lignes qui suivent sont donc plutôt des réflexions d'ordre personnel sur la façon dont un enseignant parmi d'autres envisage sa position.

L'enseignement de l'informatique musicale me semble caractérisé par un style pédagogique assez particulier. Dans l'enseignement scientifique traditionnel, la démarche d'apprentissage apparaît par bien des aspects identique à la démarche de recherche scientifique, utilisant les mêmes voies, et demandant le même type d'efforts. A ceci près : on ne demande pas à l'étudiant de trouver, mais de re-trouver ce que d'autres ont déjà découvert avant lui. Le but du chercheur scientifique est clair : montrer à d'autres chercheurs un résultat original, obtenu souvent avec du matériel de laboratoire perfectionné. L'effort de l'étudiant - comme celui de l'enfant - débouchera sur tout autre chose : il montrera par exemple à un enseignant qu'il sait démontrer un théorème. Mais dans tout cela, ni collègues, ni matériel, ni surtout de plaisir. L'enseignement scientifique réclame trop souvent chez l'étudiant l'effort préparatoire au plaisir, tout en l'excluant du plaisir de trouver.

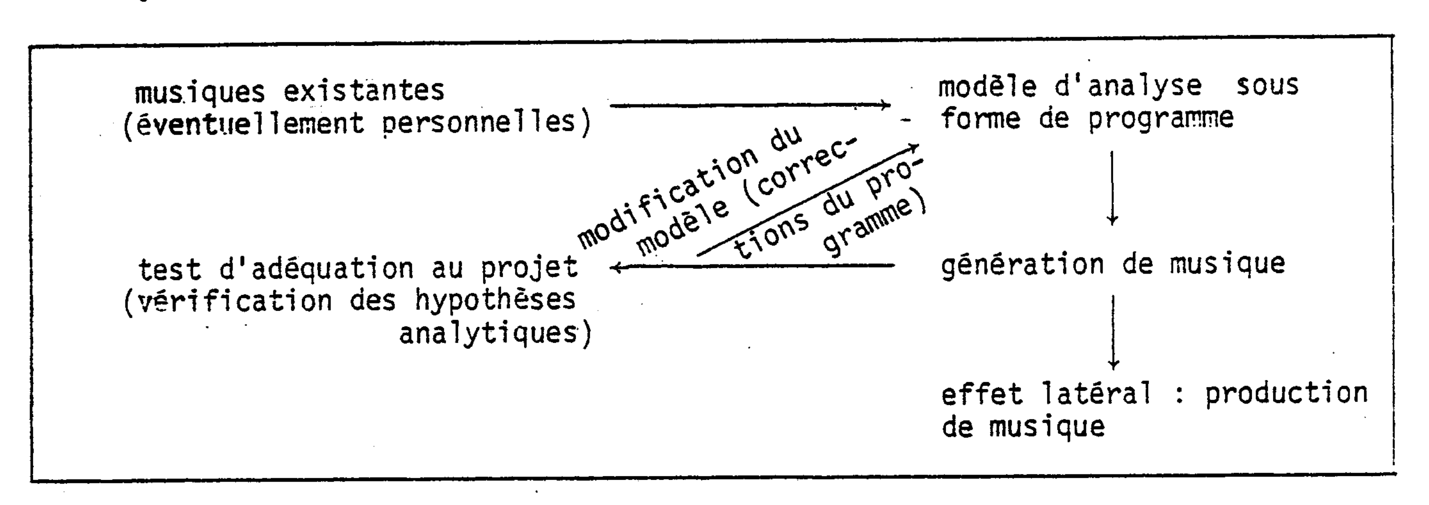
L'utilisation de périphériques musicaux ou visuels permet de donner une appréhension certes partielle, mais en tout cas intuitive, immédiate, sensible de concepts abstraits transmis habituellement dans l'enseignement des mathématiques, de la logique ou de la physique. La maîtrise d'un concept nouveau y renvoie par conséquent d'abord à un plaisir, avant de renvoyer à l'institution enseignante. C'est à partir de ce plaisir que l'étudiant désirera maîtriser d'autres concepts.

Le style de pédagogie que nous développons vise donc à favoriser un mode de communication entre étudiants proche de celui qui lie deux chercheurs. Un glissement comparable semble modifier aussi le rapport maître-élève.

Dans un contexte logico-mathématique habituel, il est normal de penser qu'il existe de bonnes solutions à un problème, et que certaines solutions sont meilleures que d'autres. En programmation usuelle, on peut dire qu'en fonction d'un but déterminé d'avance, il existe des programmes plus performants que d'autres. La programmation de compositions musicales est toute différente : il est licite de modifier en cours d'élaboration le but fixé à l'avance, et les compositeurs ne s'en privent pas. Peut-on établir dès lors une hiérarchie des itinéraires, lorsqu'on ne connaît pas d'avance le but du voyage? Les bonnes solutions deviennent caduques. Et lorsqu'elles persistent, elles prennent souvent un tour

largement empirique. On comprend qu'enseignant et enseigné entretiennent volontiers des rapports différents du modèle habituel.
D'autres facteurs viennent encore y concourir : la machine est un juge sans passion pour dire qu'un programme tourne (ce qui permet à l'enseignant de faire l'économie de certaines appréciations subjectives) ; l'enseignant et l'étudiant prennent les mêmes risques en proposant leur musique à des auditeurs ; être compositeur enfin n'est pas un grade universitaire, ce qui entraîne un décentrement des processus identificatoires habituellement mis en jeu à l'université.

Le parcours pédagogique proposé aux étudiants suggère de mêler chaque fois que c'est possible le *faire* et le *se voir faire*. L'informatique a ceci de particulier qu'elle permet une reversibilité immédiate des modèles d'analyse : l'analyse d'une pièce musicale sous forme de programme, aussi approximatif et incomplet que soit le décryptage des processus compositionnels supposés la sous-tendre, permet l'utilisation immédiate de ce programme "en génération", pour confirmation ou infirmation des hypothèses analytiques formulées. Mais dans l'un et l'autre cas de la musique est produite par effet latéral, musique écoutable en fonction de critères autres que ceux qui ont trait à l'activité d'analyse :



Est-il risqué de supposer qu'il puisse exister ainsi un jeu d'allers et retours entre le Vrai et le Beau? Est-il risqué d'avancer qu'une pièce musicale réussie puisse s'originer dans une analyse invalide? Une erreur n'est jamais fortuite. Une erreur dans l'analyse n'est-elle pas, dans certains cas, ce qui justement appartient en propre à l'analyste, et la marque de la subjectivité du compositeur?

Laissant ces questions ouvertes, il faut quand même insister sur la notion de fin. J'aimerais dire qu'un programme de composition n'est jamais fini, que sa réalisation n'est qu'une suite d'états entrecoupés de modifications et de corrections d'erreurs, corrections toujours significatives par rapport à la représentation que se fait un compositeur de sa propre pièce en cours, par rapport à ce qu'il sait de lui-même.

Ceci débouche sur un des points que je considère comme important pour les étudiants : écrire des programmes compositionnels est certes un moyen d'apprendre sur la musique, mais aussi un moyen d'apprendre sur soi-même en train de faire de la musique. J'envisage volontiers les

programmes

à composer (mais aussi : à lire, à improviser) de la musique comme des programmes d'Intelligence Artificielle, et j'essaye de faire travailler les étudiants dans la perspective que composer par programme, c'est toujours pour une part se simuler soi-même en train de composer.

★
★ ★

Venons-en aux difficultés, en examinant dans un premier temps celles qui sont directement liées à la pratique musicale avec ordinateurs.

Il faut d'abord reconnaître qu'il existe un hiatus entre le style de travail du compositeur et le style de travail du programmeur. Le compositeur traditionnel peut mener un travail isolé, visant à un résultat unique (une pièce particulière). Le processus compositionnel est contrôlé de bout en bout par une seule personne.

Le travail informatique impose une très forte dépendance par rapport à la machine (au niveau des pannes par exemple), par rapport aussi à un certain nombre d'autres personnes de technicités différentes. Mais ce qui est plus important peut être la perte de contrôle par le compositeur du processus qu'il a lui-même engagé : un programme est, on le sait, la définition d'un processus général, aux virtualités telles que tous les résultats possibles ne sont pas explorables exhaustivement ni représentables a priori, susceptible par conséquent de donner dans un cas précis des résultats inattendus ou musicalement inacceptables.

Sauf à adopter la position de principe d'accepter les yeux fermés ce qui advient une fois le programme lancé, cette perte de prévisibilité sur le résultat musical me semble à l'origine de l'échec de certains compositeurs déjà formés dans leurs tentatives de passage à la composition automatique.

Les rapports sont aussi parfois un peu difficiles entre les musiciens et ceux qui les entourent : scientifiques (mathématiciens, physiciens) et techniciens (électroniciens, programmeurs "purs").

On assiste parfois à un dialogue de sourds entre le musicien et le scientifique, celui-ci ayant tendance à indexer un jugement esthétique sur une argumentation touchant à la trivialité mathématique des algorithmes mis en jeu dans la composition. Un musicien le supporte mal : pour lui seul le résultat musical compte.

La collaboration avec le technicien a tendance à prendre dans certains cas un tour vaguement persécutif : sentiment chez le technicien d'avoir fait la plus grande partie du travail sans pouvoir en tirer le bénéfices publics, de voir un autre se contenter de signer pour recueillir les applaudissements. Le technicien, ou réputé tel, peut alors adopter une position ambivalente : réaliser du *hardware* ou du *software* à vocation musicale sans l'utiliser en son propre nom (puisqu'il ne se revendique pas comme artiste et ne se sent pas armé pour le concert) tout en empêchant plus ou moins consciemment de musiciens de l'employer (sa créativité refoulée s'incarnant alors dans des modifications "hard" ou "soft" deux jours avant le concert, pour le plus grand dépit du musicien

De son côté, le musicien pourra se sentir en risque d'être dépossédé de son objet propre à un double niveau : au niveau conceptuel par le scientifique, au niveau technologique par le technicien. Ce sentiment - un peu magique - de dépossession n'est d'ailleurs pas nouveau. On l'avait déjà vu opérer pour la musique électronique, qui témoignait d'une mutation : la lutherie n'était plus affaire artisanale, mais branche privilégiée de l'industrie, avec tout ce que ça comporte de

rentabilisation et d'amortissement commercial des matériels nouveaux.

Pour éviter ces écueils, dont j'exagère sans doute l'importance en dramatisant un peu le propos, j'insiste sur la nécessité que les étudiants en musique acquièrent une relative indépendance : il ne s'agit pas d'être omniscient ou omnicompétent, mais que le niveau informatique du compositeur soit suffisant pour lui permettre d'exprimer avec souplesse et précision des impératifs compositionnels qui lui sont tellement personnels qu'ils risquent de ne pas pouvoir être communiqués à autrui sans malentendu.

Mais les rapports interhumains les plus difficiles sont certainement ceux que l'on entretient avec soi-même. Le travail compositionnel avec ordinateur me semble particulièrement exposé à une certaine fuite en avant : les programmes réalisés ne sont pas communiqués, car on n'a pas le temps de les annoter . Le temps manque pour en donner une version commentée et lisible car on est déjà pris par la réalisation d'un nouveau programme. Résultat : tout un *software* passionnant, fruit d'une dépense d'énergie passionnée, risque de n'être finalement utilisé que par celui qui l'a conçu.

Plus : le programme aux virtualités les plus générales n'est souvent utilisé qu'une seule fois, pour une pièce particulière, par celui qui l'a écrit. Dans certains cas, cela débouche même sur un véritable art conceptuel : le programme musical ne produit pas même une seule fois de la musique. Il a tourné, c'est suffisant, et on passe à un autre programme. On a l'impression par moments que l'effort mis en jeu pour l'élaboration d'un programme est démesuré par rapport à l'exploitation parcimonieuse (ou absente) qui en sera faite.

L'informatique augmente encore chez certains ce risque connu : à force de se décrire soi-même en train de faire de la musique, avec tout ce que cela comporte de complaisance narcissique, on ne fait plus du tout de musique. La fabrication d'un *software* débouche sur la conception d'un autre *software* et la production proprement musicale disparaît au profit de la confection de programmes qui ne seront jamais exploités.

Comment, à l'université, où la fonction de communication doit tenir la première place, et où l'on n'est pas toujours astreint à justifier son travail par des traces écrites, parer au danger d'une activité tendant vers la récursion sans condition d'arrêt ni effet latéral?

Nous insistons à Vincennes, sur le fait qu'un programme musical, qu'il soit écrit par un enseignant ou un étudiant, doit d'une part être communiqué, doit déboucher aussi sur une musique audible. C'est pourquoi je donne tant d'importance à l'organisation d'auditions et de concerts, et à l'existence de recueils écrits de programmes commentés, quel qu'en soit le niveau. C'est en ce sens que nous avons créé un périodique à parution aléatoire, voué à l'informatique artistique : ARTINFO/MUSINFO.

*

* *

Second type de difficulté : celles qui sont liées à l'implantation universitaire.

Contentons nous de citer au passage une question qui se pose à chacun : va-t-on payer une activité aussi onéreuse par rapport à ses maigres fruits? Et aussi : qui va payer? Car une administration conçoit

mal qu'une matière réputée littéraire ou artistique réclame un budget qui l'aligne plutôt sur les disciplines scientifiques.

Rappelons simplement qu'une université devrait pouvoir tirer une notoriété des concerts donnés sous son nom, comme elle fait avec les articles scientifiques qui y sont réalisés.

Un aspect très positif de l'université reste néanmoins qu'elle permet à des compositeurs de vivre en se livrant à leur recherche personnelle. Mais la pratique enseignante n'est pas sans contraintes : il faut satisfaire dans le même temps des étudiants en musique débutant en informatique, et des étudiants en informatique non-musiciens (et non-susceptibles de recevoir un apprentissage musical de base, que l'université n'est d'ailleurs pas là pour assurer).

J'essaye de contourner ce premier écueil en proposant autant que possible des voies d'accès diversifiées à l'informatique musicale. A l'étudiant en musique, souvent déjà muni d'un projet musical, on conseillera très vite de mettre en chantier un programme personnel, et d'apprendre autant que possible la programmation "sur le tas", au fur et à mesure de ses besoins, *mutatis mutandis*. On insistera aussi sur l'analyse de programmes existants, propre à montrer quel types d'algorithmes ont été mis en jeu par des compositeurs devant telle ou telle nécessité compositionnelle. L'étudiant en musique se constituera ainsi son propre stock d'algorithmes, en partant du musical. Avec l'étudiant en informatique, l'accent sera mis sur l'écoute de bandes magnétiques, et sur la manipulation concrète du son : étant déjà familiarisé avec les organes de calcul, il lui faut découvrir les périphériques musicaux et apprendre à en *jouer*. L'approche ludique permet dans bien des cas de contourner des sentiments d'impuissance.

Encore une chose : la vocation primordiale de l'université est de former d'une part des enseignants en musique pour les écoles secondaires, d'autre part des informaticiens, en tout cas pas des compositeurs. Ce type de contradiction, peu gênant avec la majorité des étudiants, est susceptible de le devenir chez ceux des musiciens pour qui la programmation musicale tourne à la monomanie. On leur conseille généralement de passer deux licences, en musique et en informatique, qui leur assureront une subsistance de programmeur tout en leur permettant de se livrer impunément à leur obsession favorite!

★
★ ★

On finira sur un troisième ordre de difficultés propres à notre pratique : celles qui concernent directement la pédagogie. J'y distinguerai trois volets : ce que doit affronter l'étudiant en musique, ce que doit surmonter l'étudiant en informatique, pour conclure avec les tracas personnels de l'enseignant.

Les étudiants en musique se montrent en général très rapides dans l'apprentissage de la programmation : ils sont en effet habitués à la manipulation symbolique, familiers des processus temporels, munis d'un projet propre. Mais ils souffrent souvent d'un grave défaut : ils veulent aller trop vite, et s'exposent de ce fait à ce qu'on pourrait appeler "la dépression du deuxième mois". Munis de références musicales illustres

et de projets somptueux, ils sont déçus par l'extrême pauvreté musicale qui résulte d'une programmation encore balbutiante.

La thérapeutique préventive de ce genre d'accès dépressifs réside à mon avis dans la confection de système pédagogiques spécialisés, qui permettent de manipuler un matériau musical intéressant malgré des connaissances encore rudimentaires. C'est en ce sens qu'on été conçus à Vincennes des systèmes (encore trop peu nombreux) tels que KRWTH (J. Chailloux) ou BISEQUENCEUR (P. Greussay).

Autre danger : croire que leur savoir musical implicite, qui est considérable, va être pris en compte par la machine. Pour faire découvrir que l'ordinateur ne sait que ce qu'on lui fait connaître, j'insiste toujours avec le débutant sur la formulation en langue naturelle son propre savoir sur un point particulier, avant d'en organiser une représentation informatique.

Une autre difficulté pour certains, qui s'incarne par la confection de programmes démunis de données : arriver à concevoir qu'un programme définit un processus général, renouvelable à chaque exécution. L'idée de paramétrage d'une pièce de musique est en effet difficile pour certains musiciens.

On se trouve enfin, assez rarement il est vrai, devant un blocage de certains musiciens face à la programmation, ou de résistance à la programmation ("Je comprends, mais je ne peux pas"). Un tel phénomène me paraît souvent lié à l'impossibilité de formaliser, donc de réduire un problème musical. Comment préserver en effet ses utopies musicales personnelles si on commence à les couper en tranches, et à ramener à des automates finis ce qui n'y est pas réductible?

Ce genre de blocage me semble susceptible d'être évité si l'étudiant est mis en contact dès le premier jour avec la machine, et fait tourner son premier programme à l'issue du premier cours. Il faut s'amuser avec la machine sur un mode dédramatisé, sans laisser s'installer un décalage entre savoir théorique et pratique de la programmation, qui n'irait qu'en s'aggravant. La confection de recueils d'initiation, où tout exercice, le plus simple soit-il, est à pretexte musical s'est montrée dans ce domaine très utile.

Quant au risque d'auto-contemplation par auto-simulation évoqué plus haut, il ne résiste habituellement pas à l'existence de périphériques sonores ou visuels, qui donnent une limite dans le réel, et amènent à être vu ou entendu par un public.

L'étudiant en informatique est confronté à des difficultés d'un tout autre ordre : hormis le cas de celui qui est là pour être crédité d'une assiduité, et dont il convient de vaincre la passivité, on pourrait les résumer par ces phrases cent fois entendues :

- "Je ne suis pas artiste"
- "Je n'y connais rien en musique"
- "Je n'ai pas d'idées musicales à programmer".

L'enseignant doit alors s'efforcer de développer un certain nombre de points de vue. C'est quelquefois fort long et demande dans certains cas de faire un gros effort de persuasion :

- il faut arriver à prouver à chacun qu'il porte en lui de la musique, et que son savoir musical transmis oralement est important
- il faut vaincre les problèmes liés à la notation : montrer que la notation traditionnelle n'est ni exhaustive ni absolue, que la notation musicale ne rend compte que d'une partie de

la musique qui en sera tirée (toute notation n'est que partielle), qu'une notation rudimentaire est susceptible d'engendrer des effets audibles importants, qu'on peut inventer enfin une notation musicale pour ses propres besoins. Tout ceci se prouvera sur des instruments acoustiques, avec des exemples audibles.

Ces difficultés sont encore majorées chez des étudiants du tiers-monde, nombreux à Vincennes, que l'on doit aider à se débarasser de l'idée (post-coloniale) du caractère universel et absolu de la musique de tradition occidentale, et de sa notation.

J'ai été souvent aidé en conseillant aux étudiants de travailler en paire (un étudiant en musique, un en informatique) et en préconisant le détournement des algorithmes : on peut souvent trouver des applications musicales à des programmes non musicaux déjà existants et bien contrôlés par l'étudiant en informatique.

Je finirai ce trop long exposé en évoquant quelques unes des difficultés que j'ai rencontrées en tant qu'enseignant.

Il est d'abord difficile d'être toujours intéressant : on n'est intéressant que si l'on parle avec passion, et l'on est passionné que par ce qu'on est en train de faire. Mais l'informatique musicale est récente et va très vite. On ne saurait y fournir à l'étudiant des "bases", sorte de savoir collectif commun à tous ses praticiens. On ne peut donc enseigner qu'un savoir relié aux recherches courantes, en sachant qu'on ne peut les totaliser et que ses propres spéculations sortent parfois du champ d'intérêt des élèves. C'est dire l'importance de dispositifs ouverts sur la recherche internationale : bibliothèque, abonnements à des revues, prêt d'articles, copies de bandes. Car même si l'enseignant est en retard sur le progrès, il faut que l'étudiant puisse avoir accès à ce qui se passe ailleurs, et qui peut répondre à ses préoccupations personnelles.

Encore quelques indications, en vrac : quel langage de programmation? Toujours deux simultanément à mon sens : un langage évolué (à Vincennes LISP ou FORTRAN) et un assembleur, en s'appuyant sur un recueil d'exercices multilingues : chaque exercice est écrit dans plusieurs langages. Quel contenu donner au cours? Etre aussi varié que possible : programmation théorique et pratique, analyse de programmes et d'oeuvres, écoute de musique, panorama international. Sans oublier l'abord historique que les étudiants apprécient toujours jusque dans l'anecdote : quelle est la place de la *computer music* dans la musique contemporaine? Quelle est la place de tel compositeur dans la *computer music*?

Je soumets enfin au lecteur deux questions que je tiens à laisser ouvertes, et qui touchent à la spécificité ou non de la musique à l'ordinateur :

- l'analyse d'une oeuvre composée avec ordinateur peut-elle se réduire à l'analyse du programme qui l'a engendrée?
- doit-on enseigner la composition musicale avec ordinateur comme une discipline indépendante ? Autrement dit : les méthodes traditionnelles de composition musicale sont-elles une base nécessaire, ou au contraire un fardeau inutile pour le compositeur qui travaille sur ordinateur?

Arrivé à la coda, je m'aperçois qu'il est nécessaire de modifier mon incipit. J'y remerciais un ami, mais il convient avant tout de remercier les étudiants de Vincennes. Car lui et moi leur sommes redevables, directement ou indirectement, des quelques idées exprimées dans cette communication.